IMAGE BLOCK CLASSIFICATION BASED ON ENTROPY OF DIFFERENCES

BACKGROUND

[0001] The present invention relates to digital images. More specifically, the present invention relates to edge detection in blocks of digital images.

[0002] Compound documents may contain text, drawings and photo regions (sometimes overlaid), complex backgrounds (e.g., text boxes), watermarks and gradients. For example, magazines, journals and textbooks usually contain two or more of these features.

[0003] A single compression algorithm is usually not suitable for compressing compound documents. Compression algorithms such as JPEG are suitable for compressing photo regions of the compound color documents, but they are not suitable for not compressing text regions of the compound color documents and other regions containing edges. These lossy compression algorithms are based on linear transforms (e.g., discrete cosine transform, discrete wavelet transform) and do not compress edges efficiently. They require too many bits, and may produce very objectionable artifacts around text.

[0004] Compression algorithms such as CCITT, G4 and JBIG are suitable for compressing black and white text regions and other regions containing edges. However, compression algorithms such as CCITT, G4 and JBIG are not suitable for compressing photo regions.

[0005] A typical solution is to pre-process the documents, separating the regions according to the type of information they contain. For instance, regions containing edges (e.g., regions containing text, line-art, graphics) and regions containing natural features (e.g., regions containing photos, color backgrounds and gradients) are separated and compressed according to different algorithms.

[0006] However, algorithms for separating the regions tend to be very complex, requiring large amounts of memory and high bandwidth. The complexity, high bandwidth and large memory requirements make many algorithms unsuitable for embedded applications such as printers, scanners and

other hardcopy devices.

. SUMMARY

[0007] According to one aspect of the present invention, edges in a block of a digital image are detected by determining an entropy of differences in pixel values. Other aspects and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figure 1 is an illustration of a method for detecting the presence of edges in an image block.

[0009] Figures 2a and 2b are histograms representing natural image features in an image block.

[0010] Figures 2c and 2d are histograms representing text and graphics in an image block.

[0011] Figure 3 is a histogram representing an exemplary image block containing a vertical edge.

[0012] Figure 4 is an illustration of a method of using a lookup table to determine the entropy and maximum pixel difference in an image block

[0013] Figure 5 is an illustration of a hardware implementation of the method of Figure 1.

DETAILED DESCRIPTION

[0014] As shown in the drawings for purposes of illustration, the present invention is embodied in a method for detecting edges in blocks of a digital image. The method involves creating a histogram from differences in pixel luminance, and computing the entropy of the histogram. This method is much simpler than conventional approaches, such as morphological analysis. The entropy can be computed very efficiently using only a look-up table.

[0015] Reference is made to Figure 1, which illustrates a method for

detecting edges in a block of a digital image. The digital image is made up of a plurality of pixels, each pixel represented by an n-bit word. The edge detection method is concerned with differences in luminance; therefore, if the n-bit words represent RGB color space, they should be converted to luminance values. For example, the n-bit words could be converted to grayscale values (e.g., [R+G+B]/3) or values of a luminance component of a color space (e.g., YUV, Yab) having a luminance (Y) component.

[0016] The block may be of any size. For example, the block may by an 8x8 block of pixels. The blocks may have geometries other than square.

[0017] The edge detection does not identify the location of an edge in a block; it merely indicates the presence of an edge in a block. The edge detection method is performed by computing a histogram of absolute differences in luminance values of adjacent pixels (110). Horizontal and vertical absolute differences of adjacent pixels may be computed as follows:

$$d_{i,j} = \left| p_{i,j} - p_{i,j-1} \right| \quad \text{(horizontal)}$$

$$d_{i,j} = \left| p_{i,j} - p_{i-1,j} \right| \quad \text{(vertical)}$$

where and $p_{i,j}$ represents the luminance value of the pixel in the ith row and j^{th} column of the block.

[0018] Figures 2a and 2b show exemplary histograms of natural features (e.g., photos), graphic arts and complex patterns. Figures 2c and 2d show exemplary histograms of text and graphics. Blocks that are part of photos or complex graphic patterns usually have histograms that are flat or are peaked at zero (flat regions on photos). Blocks with text and graphics normally usually have histograms containing a few isolated peaks. Sometimes the edges are blurred, producing peaks that are not isolated, plus some random differences.

[0019] The entropy of the histogram is computed to determine whether edges are present in the block (112). The entropy may be computed as follows.

$$E(h) = \log(T) - \frac{1}{T} \sum_{h_n \neq 0} h_n \log(h_n)$$

where n = 0, ..., C are a set of non-negative numbers corresponding to bin numbers of the histogram; h_n = 0, ..., C are a set of non-negative numbers

-3-

corresponding to heights (i.e., frequencies of occurrences) of the bins; and

$T = \sum_{n=0}^{C} h_n$ (the total area of the histogram).

[0020] Consider the example of a 4x4 block having the following luminance values.

0	0	255	255
0	0	255	255
0	0	255	255
0	0	255	255

These values represent a vertical edge. Horizontal and vertical absolute differences computed according to the equations above yield the following results: 12 vertical 0's, 8 horizontal 0's and 4 horizontal 255's. Thus the histogram, which is shown in Figure 3, has two non-zero bins: $h_0=20$ and $h_{255}=4$. All other bins (h_1, ..., h_{254}) equal zero. The total area T=24, and the entropy E(h) equals

$$E(h) = \log(24) - \frac{1}{24}[20\log(20) + 4\log(4)].$$

After the entropy value E(h) has been determined, it is compared to an absolute threshold (114). The maximum absolute difference in the block is also determined (116) and compared to a threshold (118). These comparisons indicate whether the block contains a significant edge. Smooth regions are characterized by low entropy and a low maximum difference. Regions having large random differences (e.g., noise) are characterized by high entropy and a high maximum difference. Regions having edges are characterized by low entropy and a high maximum difference. In the example above, the entropy is low and the maximum difference (255) is high. Therefore, the block is identified as containing an edge. The location of the edge and the type of edge (horizontal, vertical) are not determined; only the presence is determined.

[0021] The maximum difference is a simple way of determining whether the block contains a smooth region or an edge. Variance is a different measure that can be used to determine whether the block contains a smooth region or an edge.

-4-

[0022]   After a block is processed, it may compressed by the appropriate compression algorithm.   The edge detection allows the proper compression algorithm to be applied to the block. If a block contains an edge, it is encoded with a lossless compression algorithm. If a block does not contain an edge, it is encoded with a lossy compression algorithm.

[0023]   Since the entropy function E(h) is computed from logarithms and multiplication, it would appear to be computationally costly. However, it is quite the opposite. Factors of the entropy function can be pre-computed, scaled and rounded to integers, so that all complex computations are replaced by table look-up.

[0024]   The entropy function may be normalized to allow a single comparison to be performed (instead of a first comparison to an entropy threshold and a second comparison to a maximum difference threshold).   The entropy function may be normalized as follows:

$$E'(h) = \frac{a(E(h)+b)}{\mu(h)+c}$$

where E'(h) is the normalized function, μ(h) is the maximum argument in the histogram, and a, b, and c are constants that normalize the entropy function E(h), and change the sensitivity of the classification. The maximum argument μ(h) may be computed as $\mu(h) = \max_{h_n \neq 0}(n)$. The constants should be chosen so that E'(h) is small when the block contains edges, and E'(h) is large in all other cases. Under those conditions, a block is classified as containing at least one edge if E'(h) is smaller than a given threshold (e.g., E'(h)<1).

[0025]   Reference is now made to Figure 4. The lookup table may be used as follows. The table has Q entries (0, ..., Q-1). Each entry Q corresponds to a scaled bin height. The entries are rounded off to integers.

[0026]   Horizontal and vertical absolute differences in luminance between adjacent pixels are computed (210) and a histogram is created from the differences (212). The maximum difference is updated as the histogram is being created (214).

[0027]   After the histogram has been created, the maximum luminance

-5-

difference is compared to a first threshold T1 (216). If the maximum difference is less than the threshold, the block is identified as not containing any edges (218). If the maximum difference is greater than the threshold, the entropy of the histogram is computed.

[0028]   The entropies of the histogram bins are looked up in the table as a function of bin height (220), and the bin entropies are summed (222). If the sum is greater than a second threshold T2 (224), the block is identified as not containing any edges (218). If the sum is less than or equal to the threshold, the block is identified as containing at least one edge (226).

[0029]   Reference is now made to Figure 5, which shows a hardware implementation of the edge detection method. A block of data is stored in a buffer 310. Memory 312 stores a program for instructing a processor 314 to detect edges in the buffered block, the edges being detected according to the edge detection method herein. The processor 314 may also perform compression using an algorithm that is selected according to the results of the edge detection.

[0030]   The present invention is not limited to image compression. For example, the present invention may be applied to video and photo segmentation.

[0031]   The present invention is not limited to the specific embodiments described and illustrated above. Instead, the present invention is construed according to the claims that follow.